



Resumen sobre el patrón MVC

Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas a este nuevo video sobre servlet. En este caso, quería hacer un poco un resumen sobre lo que hemos estado aprendiendo sobre nuestro patrón MVC, nuestro Model View Controller.

[00:17] Bueno, primero tienen que saber que, tienen que recordar en realidad porque ustedes ya saben, que nosotros tenemos un servlet, un único servlet, que es quien recibe nuestras peticiones HTTP. Toda petición HTTP va a ir sí o sí hacia ese único servlet. Nunca vamos a tener una petición, por ejemplo, directamente a la view.

[00:40] Recuerden que hicimos esa modificación de nuestro JSP FormNuevaEmpresa, por ejemplo, que los sacamos de ahí y creamos una nueva action para él. De hecho, no la agregué acá, me olvidé, pero bueno, siempre va a nuestra entrada servlet.

[01:01] Y acá recuerden que dentro de nuestro controlador también están nuestras actions, todas nuestras actions son parte del controlador, no se confundan. Nuestro controlador tiene la tarea de decidir a cuál action tiene que llamar, dependiendo de cuál es la entrada que llegó, de nuestra petición HTTP. Entonces ve cuál es esa entrada HTTP y dice ¿a cuál action tengo que llamar?

[01:28] Tengo que llamar a listaEmpresas, por ejemplo, perfecto. Entonces va a derivar la tarea a listaEmpresas. Bien, ese listaEmpresas o controlador tenía

tarea de decir cuál action llamar, este listaEempresas, nuestras actions, lo van a ver como actions dentro de proyectos, tiene la tarea de saber cuál modelo tiene que instanciar, cuáles datos tiene que pegar.

[02:00] Tiene que instanciar la base de datos, tiene que sustancia el modelo, entonces tiene bastantes tareas por sí misma esa action. Así que retorna, devuelve los datos nuestro modelo, nuestra action tiene también la tarea de decir cuál JSP tiene que ser renderizada. No es que nuestra action va a llamar directamente a nuestro JSP, pero sí nuestra action decide y le comunica a nuestro controlador cuál es el JSP que tiene que ser renderizado.

[02:29] Y entonces, una vez que sabe eso y le envía esos datos a nuestro controlador, el controlador es quien llama al JSP y realmente renderiza los datos. Entonces vimos que también tenemos nuestro modelo, nuestro modelo es bastante simple hoy en día, bastante simple, tenemos solamente una base de datos que en realidad es una base de datos construida y que queda en memoria, digamos, no tiene persistencia de datos.

[02:56] No guarda datos para que, una vez que ha pagado el servidor y lo prendemos de nuevo en volver a tener los mismos datos que habíamos guardado, pero hicimos una simulación, también tenemos solamente una clase la cual es parte de nuestro modelo, en este caso es nuestra empresa. En proyectos van a ver que existen muchas más clases dentro de lo que es este modelo.

[03:22] Pero para nuestro proyecto ya es suficiente realmente tener una clase como modelo, y bueno, en proyectos futuros que ustedes tengan, van a ver que existen varias clases, pero con esto ya es suficiente para comprender ese concepto. Finalmente nosotros tenemos nuestra view, que ella nunca funciona sola.

[03:49] Existe un dicho que dice always linked to and action. Una view siempre tiene que estar linkada, conectada con una action. Entonces, la view sola

nunca va a proporcionar los datos que un usuario necesita. Siempre necesita de un action que lea los datos de la base de datos para poder enviar luego a nuestra view y que ahí sí nuestra view renderice y envíe toda esa información al navegador.

[04:26] Entonces siempre van a ver entonces este modelo o variaciones parecidas, así como está MVC, tenemos Model View en Android y tenemos MVP también, entonces existen variaciones, pero siempre vamos a tener esa view, ese model y el controller es quien más tiene variación.

[04:50] Pero no es mucho, no se preocupen tanto. Este controlador que nosotros tenemos acá, si no comprendieron bien el tema de reflexión, tampoco se preocupen tanto porque esa parte en particular del controlador normalmente es gerenciado por los frameworks como spring, por ejemplo, hace un buen gerenciamento y nosotros tenemos que ocuparnos más de las partes de las actions, no tanto del controlador.

[05:14] Entonces está todo bien. Si han comprendido MVC me quedo muy feliz, lo importante es esa separación de los archivos en capas. Bueno, en los videos futuros vamos a estar viendo un poco más sobre autenticación y autorización dentro de nuestra aplicación web, que es una cosa que falta. Entonces nada, nos estaremos viendo próximamente. Un abrazo y nos vemos allá.